AMENDMENTS TO THE CLAIMS

Please amend claims 1, 13-19, 21-23 and 26 as indicated below:

1. (Currently Amended) A method for converting data suitable for use on a source platform into data suitable for use on <u>one or more</u> [[a]] target ~~platform~~ <u>platforms</u>, said method comprising:

analyzing source platform code;

extracting information from said analyzed source platform code wherein said extracted information <u>represents system model information which</u> includes at least <u>one of</u> [[the]] logic, flow, user interface description, [[and]] <u>or</u> data of said source platform code;

<u>representing the extracted system model information in a generic reusable intermediate data format which is independent of a format used by the source platform code or on the one or more target platforms;</u>

<u>storing the generic reusable intermediate data format representing the extracted information; and</u>

~~defining a generic data structure and format for storing said extracted information;~~

~~storing said extracted information in said defined structure and format; and~~

transforming said <u>generic reusable intermediate data format representing the</u> extracted information into code suitable for <u>the one or more</u> [[said]] target ~~platform~~ <u>platforms</u> ~~wherein said transforming step comprises transforming said extracted information into code suitable for said target platform after said extracted information is stored in said defined structure and format.~~

2. (Original) The method of claim 1 wherein said defined structure and format is XML.

3. (Original) The method of claim 1 wherein said analyzing of said source platform code comprises:

defining a language recognition tool; and

using said defined language recognition tool to recognize elements of a program in a particular language.

4. (Original) The method of claim 3 wherein said language recognition tool is based on an EBNF programming language grammar.

5.   (Original) The method of claim 3 wherein said language recognition tool is an ANTLR language recognition tool.

6.   (Original) The method of claim 5 wherein said analyzing step further comprises:

defining a custom analysis tool that is specific to said source platform code; and

using said defined custom analysis tool to pre-process said source platform code before said extracting of information.

7.   (Original) The method of claim 5 wherein said analyzing step further comprises:

defining a custom analysis tool that is specific to said source platform code; and

using said defined custom analysis tool to post-process said source platform code after said extracting of information

8.   (Original) The method of claim 6 wherein said analyzing step further comprises:

defining a third party tool to be used for analyzing said source platform code; and

using said defined third party tool to identify elements of said source platform code that are relevant and not-relevant to said transforming of said extracted information.

9.   (Original) The method of claim 1 further comprising:

producing a report from said extracted information.

10.   (Original) The method of claim 9 further comprising:

analyzing and performing an intermediate transformation of said extracted information to assist with said report producing step.

11.   (Original) The method of claim 9 wherein said report comprises at least one of:

a user interface mock-up;

data definitions;

symbol counts;

application flow;

a generic XML report to assist in validating or verifying other complex manual migration of code from one platform to another platform; and

details of a status of migration of code from one platform to another platform for a user.

12.   (Original) The method of claim 1 wherein said transforming step comprises:

defining a set of transformation rules specific to said target platform; and

using said transformation rules in transforming said extracted information into code suitable for said target platform.

13.   (Currently Amended) A mechanism for migrating computer code from a source platform to [[a]] one or more target ~~platform~~ platforms comprising:

means for preparing source files;

means for extracting information from said prepared source files wherein said extracted information represents system model information which includes at least one of logic, flow, user interface description, or data of said source files;

~~a means for reverse engineering said prepared source files into an intermediate code;~~

means for representing the extracted system model information in a generic reusable intermediate data format which is independent of a format used by the source files or the one or more target platforms;

~~a means for transforming said intermediate code into code suitable for use on said target platform~~

means for storing the generic reusable intermediate data format representing the extracted information; and

means for transforming said generic reusable intermediate data format representing the extracted information into code suitable for the one or more target platforms.

14.   (Currently Amended) The mechanism of claim 13 further comprising:

a means for preparing reports on said ~~reverse engineered~~ prepared source files.

15.   (Currently Amended) The mechanism of claim 13 further comprising:

a means for creating transformation rules to assist with said transforming means; and

a means for inputting said transformation rules into said means for transforming said ~~intermediate code~~ generic reusable intermediate data format.

16. (Currently Amended) A computer program product having a computer readable storage medium having computer program logic recorded thereon for transforming code suitable for use on a source platform to code suitable for use on [[a]] one or more target platform platforms, the computer program product comprising:

code for analyzing source platform code;

code for extracting information from said analyzed source platform code wherein said extracted information represents system model information which includes at least one of logic, flow, user interface description, or data of said source platform code;

code for representing the extracted system model information in a generic reusable intermediate data format which is independent of a format used by the source platform code or on the one or more target platforms;

code for storing the generic reusable intermediate data format representing the extracted information; and

code for transforming said generic reusable intermediate data format representing the extracted information into code suitable for the one or more target platforms

code for extracting relevant information from said code suitable for use on said source platform wherein said relevant information includes at least the logic, flow, user interface description, and data from said code suitable for use on said source platform;

code for converting said relevant information into a generic representation of elements that reflect said relevant information of said code suitable for use on said source platform; and

code for transforming said generic representation of elements into code suitable for use on said target platform.

17. (Currently Amended) The computer program product of claim 16 further comprising:

code for optimizing said code suitable for use on said source platform code for extraction;

18. (Currently Amended) The computer program product of claim 16 further comprising:

code for generating reports based on said generic representation of elements reusable intermediate data format representing the extracted information.

19. (Currently Amended) The computer program product of claim 18 further comprising:

code for analyzing and processing said generic ~~representation of information elements~~ reusable intermediate data format representing the extracted information to assist said code for generating reports.

20. (Original) The computer program product of claim 16 further comprising:

code for generating an output file representing the code suitable for use on said target platform.

21. (Currently Amended) The computer program product of claim 16 wherein said code for transforming comprises:

code for inputting a set of transformation rules specific to said target platform; and

code for using said transformation rules to convert said generic ~~representation of elements~~ reusable intermediate data format representing the extracted information into said code suitable for use on said target platform.

22. (Currently Amended) The computer program product of claim 16 further comprising:

code for storing said generic ~~representation of elements~~ reusable intermediate data format representing the extracted information ~~that reflect said relevant information~~ of said code suitable for use on said source platform in XML format.

23. (Currently Amended) The computer program product of claim 22 further comprising:

code for generating an output file representing said generic ~~representation of elements~~ reusable intermediate data format representing the extracted information that reflect said relevant aspects of said code suitable for use on said source platform.

24. (Original) The computer program product of claim 17 wherein said code for optimizing comprises:

code for inputting system documentation of said code suitable for use on a source platform;

code for inputting said code suitable for use on said source platform; and

code for using said inputted documentation and said code suitable for use on said source platform to refine said code suitable for use on said source platform for extraction.

25. (Original) The computer program product of claim 24 wherein said code for optimizing further comprises:

code for performing customized extraction of information from said code suitable for use on said source platform.

26. (Currently Amended) A data processing system for transforming a computer program written for a source platform to a computer program written for [[a]] one or more target platform platforms comprising:

memory storing a transformation program operating to:

analyze a program operating on a source platform;

extract information from said analyzed source platform code, wherein said extracted information represents system model information which includes at least one of [[the]] logic, flow, user interface, [[and]] or data of said source platform code;

represent the extracted system model information in a generic reusable intermediate data format which is independent of a format used by the source platform or on the one or more target platforms;

store the generic reusable intermediate data format representing the extracted information; and

transform said generic reusable intermediate data format representing the extracted information into code suitable for the one or more target platforms

define a generic data structure and format;

store said extracted information in said defined generic data structure and format; and

transform said extracted information stored in said defined generic data structure and format into code suitable for said target platform; and

a processor for executing said transformation program.

27. (Original) The data processing system of claim 26 wherein said defined structure and format is XML.

7

28. (Original) The data processing system of claim 26 wherein said transformation program operates to analyze a program operating on a source platform by:

defining a language recognition tool; and

using said defined language recognition tool to recognize elements of said program operating on said source platform.

29. (Original) The data processing system of claim 28 wherein said language recognition tool is based on an EBNF programming language grammar.

30. (Original) The data processing system of claim 29 wherein said transformation program operates to further analyze a program operating on a source platform by:

defining a custom analysis tool that is specific to said program operating on said source platform; and

using said defined custom analysis tool to pre-process said program operating on said source platform before said extracting of information.

31. (Original) The data processing system of claim 30 wherein said transformation program operates to further analyze a program operating on a source platform by:

defining a third party tool to be used for analyzing said source program operating on said source platform; and

using said defined third party tool to identify elements of said source program operating on said source platform that are relevant and not-relevant to said transforming of said extracted information.

32. (Original) The data processing system of claim 26 wherein said transformation program further operates to:

produce a report from said extracted information.

33. (Original) The data processing system of claim 32 wherein said transformation program further operates to:

analyze and perform an intermediate transformation of said extracted information to assist with said report producing.

34.   (Original) The data processing system of claim 32 wherein said report comprises one or more of:

a user interface mock-up;

data definitions;

symbol counts;

application flow;

a generic XML report to assist in validating or verifying other complex manual migration of code from one platform to another platform; and

details of a status of migration of code from one platform to another platform for a user.

35.   (Original) The data processing system of claim 26 wherein said transformation program operates to transform said extracted information by:

defining a set of transformation rules specific to said target platform; and

using said transformation rules in transforming said extracted information into code suitable for said target platform.